

Teoría de las Comunicaciones

Segundo Cuatrimestre 2014

**Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Argentina**



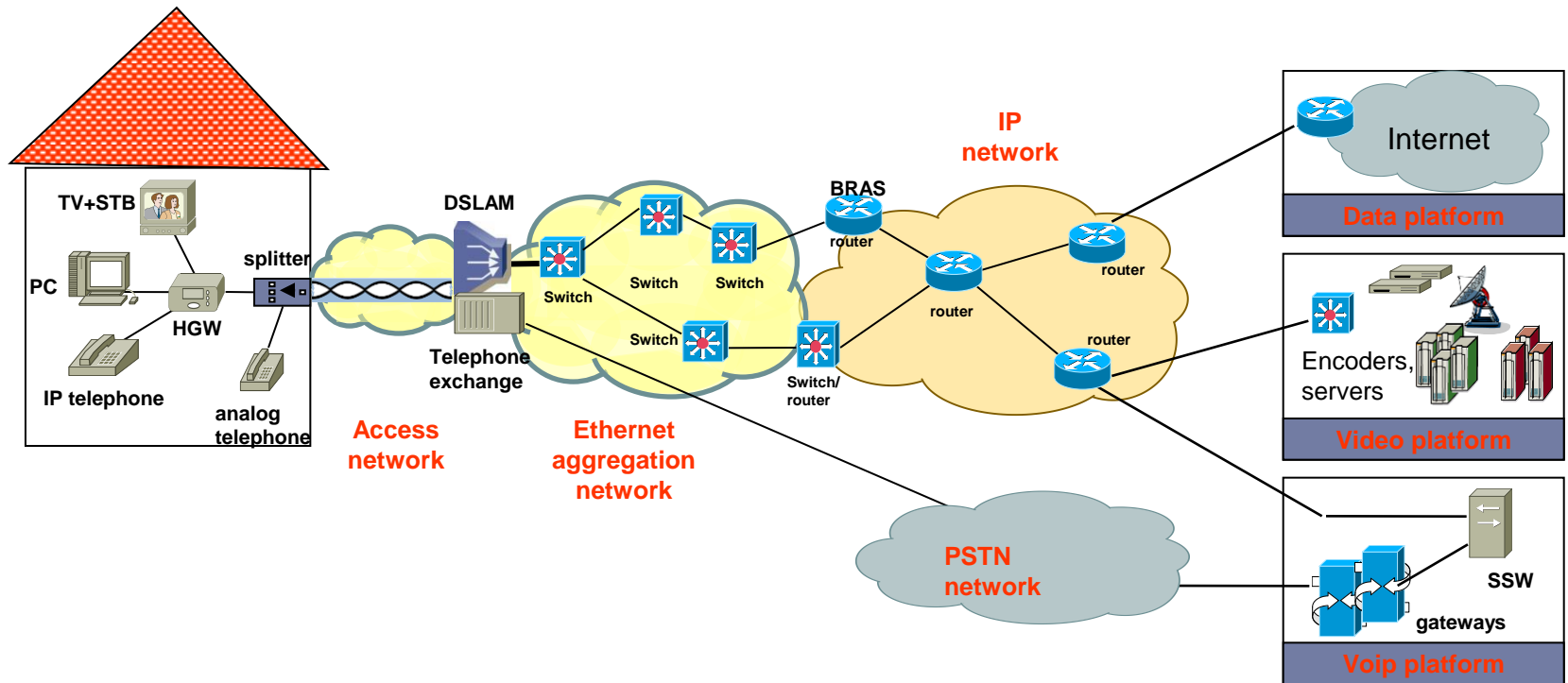
Performance



Fundamentos

Análisis de Performance

- ▶ Tenemos una notebook que accede al servicio de banda ancha ADSL de mi hogar mediante un AP 802.11g, el mismo se conecta a HGW de grafico del slide que sigue.
- ▶ El servicio ADSL que tengo lo llaman “6 Megas”
- ▶ Queremos realizar una transferencia de un archivo de 10 GB desde un servidor en USA a mi notebook.
- ▶ Cuanto estimamos que demora la transferencia ?

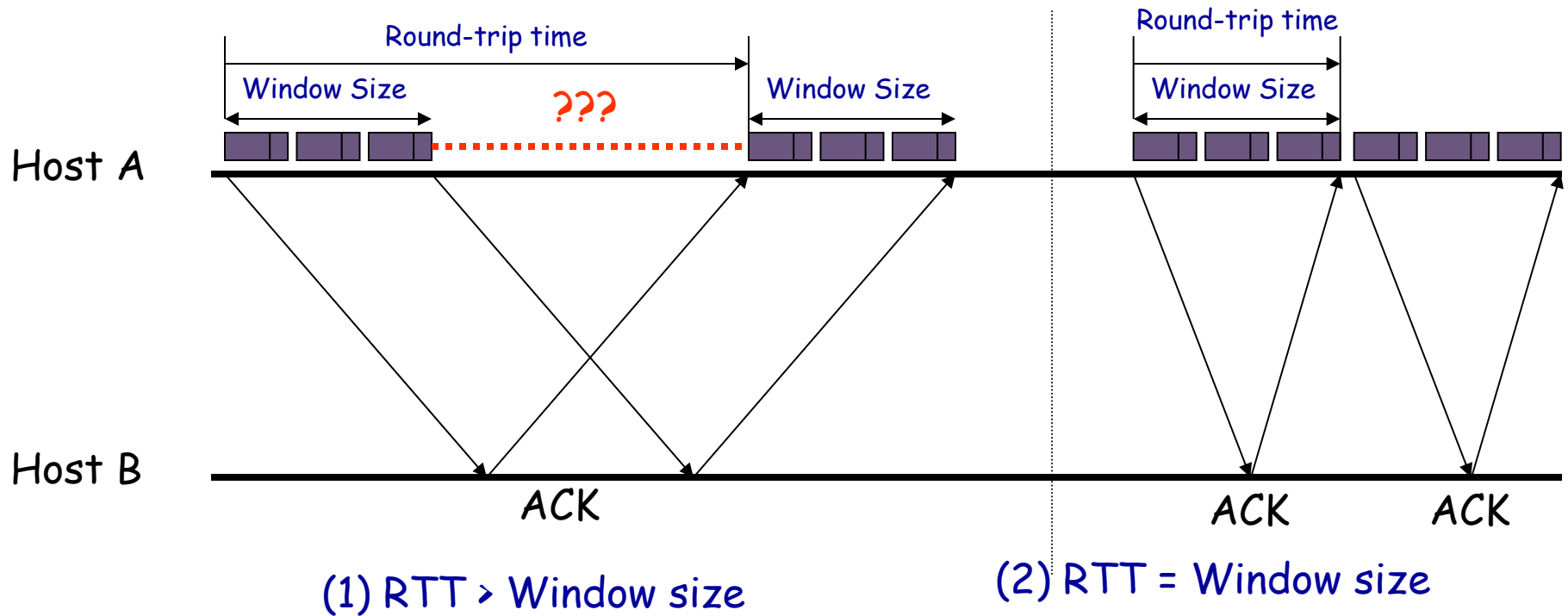


DSLAM: Digital Subscriber Line Access Multiplexer
BRAS: Broadband Remote Access Server

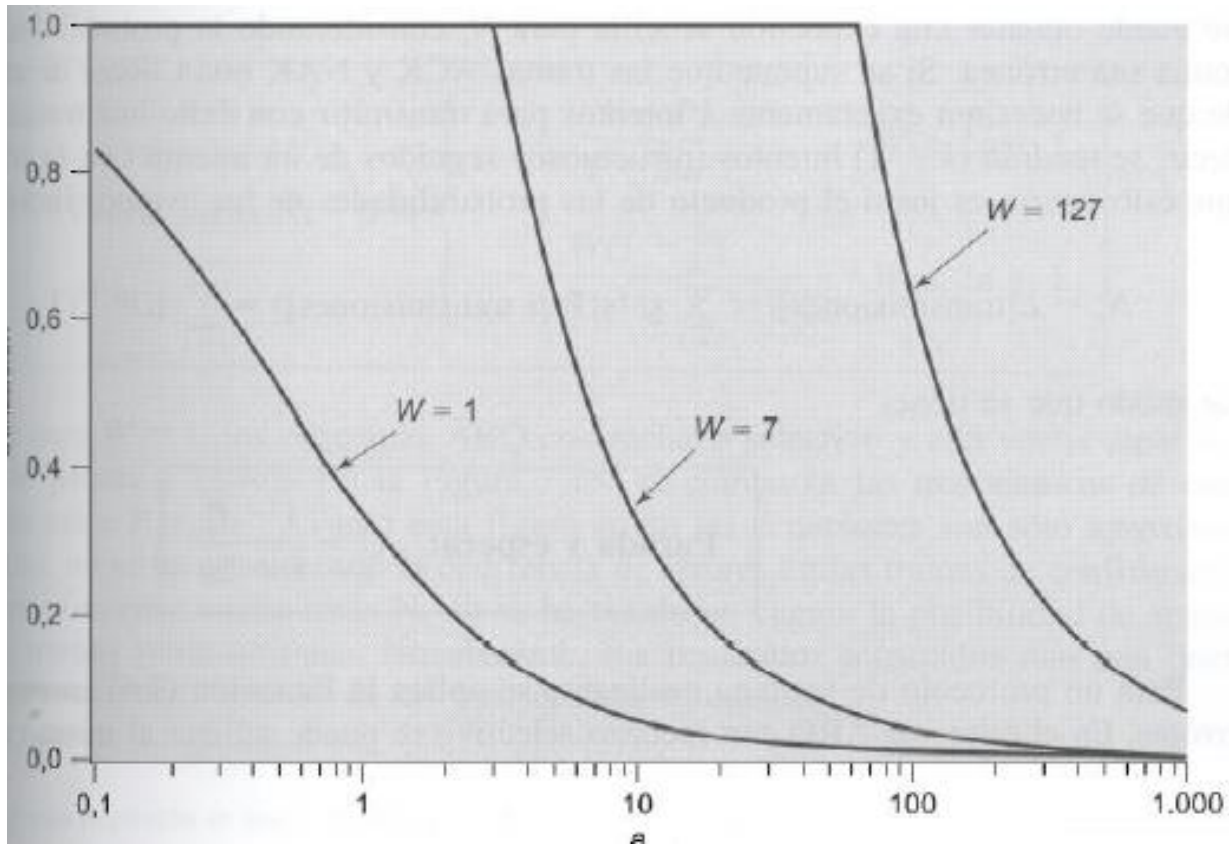
Que debemos mirar ??

- ▶ El servicio ADSL que tengo lo llaman “6 Megas”, que significa ?? Es simétrico ??
- ▶ Queremos realizar una transferencia de un archivo de 10 GB desde un servidor en USA a mi notebook.
- ▶ Cuanto estimamos que demora la transferencia ?
- ▶ Cual es RTT al servidor ?
- ▶ Tengo el upstream de mi acceso utilizado por aplicaciones P2P? En que afecta ?
- ▶ El Wi-Fi es un cuello de botella ?
- ▶ etc

Sliding Window



Utilización vs RTT normalizado



Fuente : Stallings W . Comunicaciones y Redes de Computadores- 7 Edición

Throughput protocolo ventana deslizante

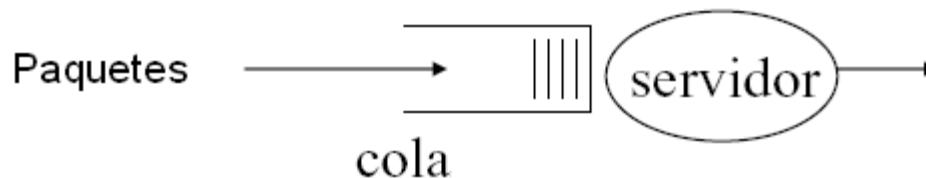
► $T < W/RTT$!!!!!!!

Perfomance del Protocolo de Ventana deslizante

- ▶ Sea un Host A, que usa protocolo de ventana deslizante, si debe transmitir un archivo de unos 10 GB con un Host B
- ▶ Window size = 64KB
- ▶ RTT de la red es de 1 segundo
- ▶ Cual es la velocidad esperada con que el emisor envia datos?
- ▶ 64Kbps

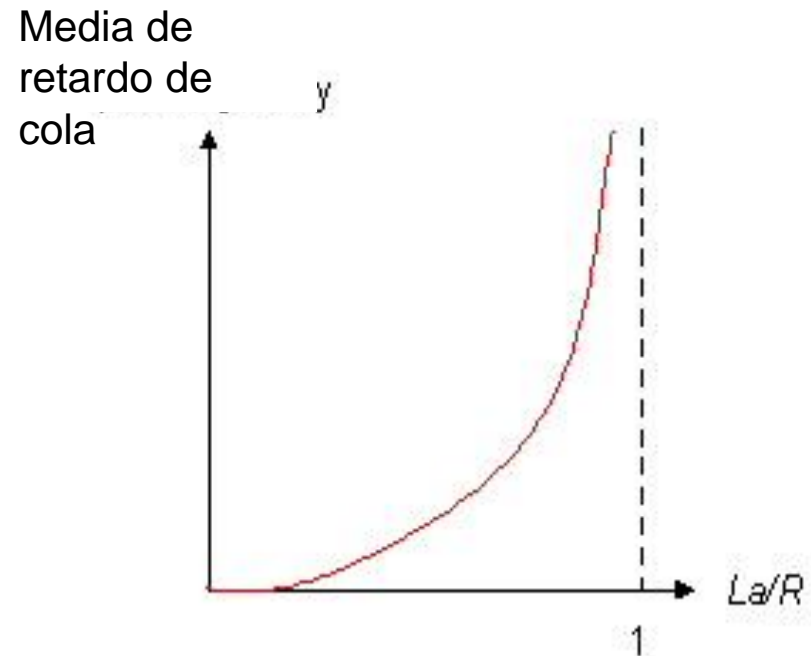
Cola M/M/1

- ▶ Es una cola en la que la entrada de los paquetes es un proceso de Poisson y el tiempo de servicio de paquetes es exponencial.
- ▶ Hay un **único servidor**.
- ▶ La cola tiene **capacidad infinita**.



Retardo Medio en una COLA – M/M/1

- ▶ R = ancho de banda del enlace (bps).
- ▶ L = longitud del paquete (bits).
- ▶ a = media de tasa de llegada del paquete.



Intensidad de tráfico = La/R

- $La/R \sim 0$: media de retardo de cola pequeño.
- $La/R \rightarrow 1$: aumentan los retardos.
- $La/R > 1$: ¡Llega más “trabajo” del que puede servirse, media de retardo infinita!

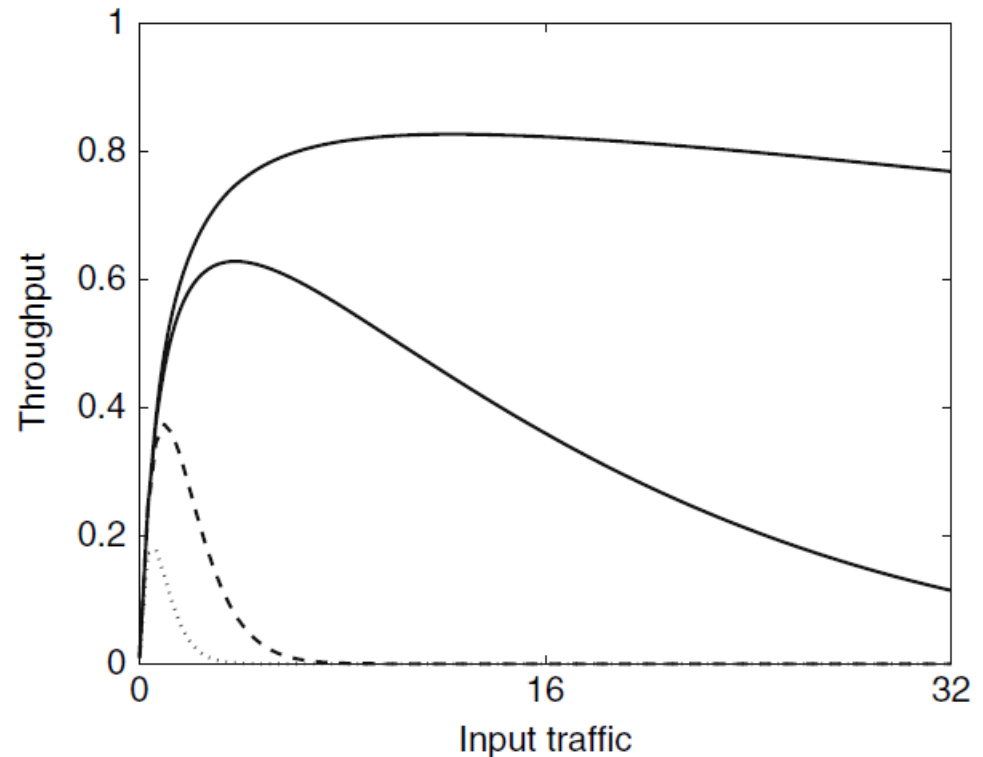
Eficiencia : raw data !!!!!

- ▶ La tabla nos presenta data rates a NI para IEEE 802.11a/g con codificación convolucional “convolutional coding.” Esos números son “raw rates” y el throughput de es de alrededor de 28 Mbps para 54Mbps (46% de eficiencia), [64QAM y 3/4 coding rate en un ancho de banda de 20-MHz].

Mode	Modulation	Code rate	Data rate (Mbps)
1	BPSK	1/2	6
2	BPSK	3/4	9
3	QPSK	1/2	12
4	QPSK	3/4	18
5	16QAM	1/2	24
6	16QAM	3/4	36
7	64QAM	2/3	48
8	64QAM	3/4	54

DCF MAC :Performance

El Throughput de IEEE 802.11/DCF versus el average input traffic cuando $w = 8$, $n = 10$, y $N = 32$ (línea superior llena) . La línea siguiente llena es el throughput de CSMA/CA, la línea rayada el throughput de slotted ALOHA, y la punteada de ALOHA puro.



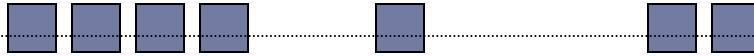
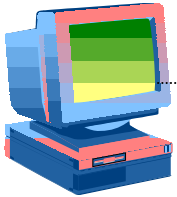
“Policing Mechanisms”

- ▶ **Average Rate** (100 paquetes por segundo o 6000 paquetes por minuto), un aspecto crucial es la longitud del intervalo
- ▶ **Peak Rate**: e.g., 6000 p p minute Avg and 1500 p p sec Peak
- ▶ **Burst Size**: número máximo de paquetes enviados consecutivamente (en un periodo corto de tiempo)

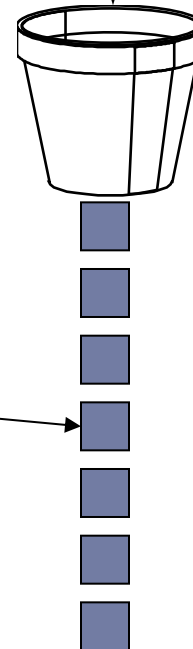
“Traffic Shaping”

- ▶ El trafico es “*bursty*” (*con ráfagas*) impacta en la congestión.
- ▶ Traffic shaping es un método de lazo abierto que trata de guiar la congestión , forzando a los paquetes a transmitirse a una velocidad mas predecible
- ▶ Se trata de mantener el trafico constante , es decir regular la tasa de transmisión media (y el “*burstiness*”) de los datos .
- ▶ Por ejemplo en las redes de CV , en el setup de circuito es usuario y el carrier se pueden de acuerdo en el conformado (shape) del circuito. Se necesita de un acuerdo del proveedor y el cliente

Algoritmo “Leaky Bucket”(idea)



Flujo de paquetes sin regular



El flujo de salida tiene una velocidad constante ρ , cuando hay agua en el balde, y cero cuando esta vacío

=> Se moderan las ráfagas

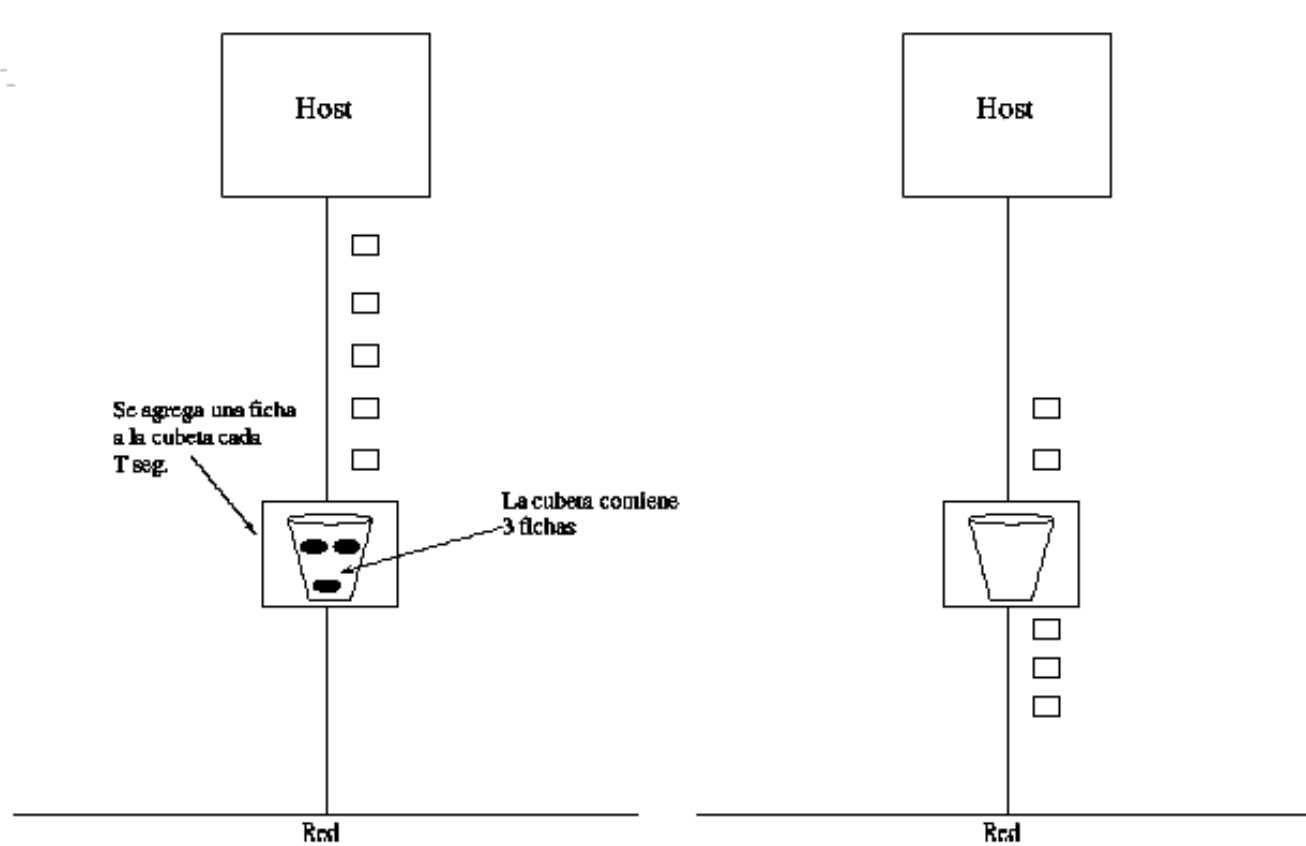
Implementación

- ▶ El mecanismo de leaky bucket no es mas que un sistema single-server queueing con tiempo de servicio constante y cola finita.
- ▶ Paquetes llegan en cualquier instante , pero a los host se le permite poner solo un paquete por “clock tick” en la red .
- ▶ Si los paquetes son de diferentes tamaños , es mejor usar un numero fijo de bytes por “tick”, antes que uno por paquete
- ▶ Cola llena, los paquetes que llega son descartados (tail drop)

Algoritmo: “Token Bucket”

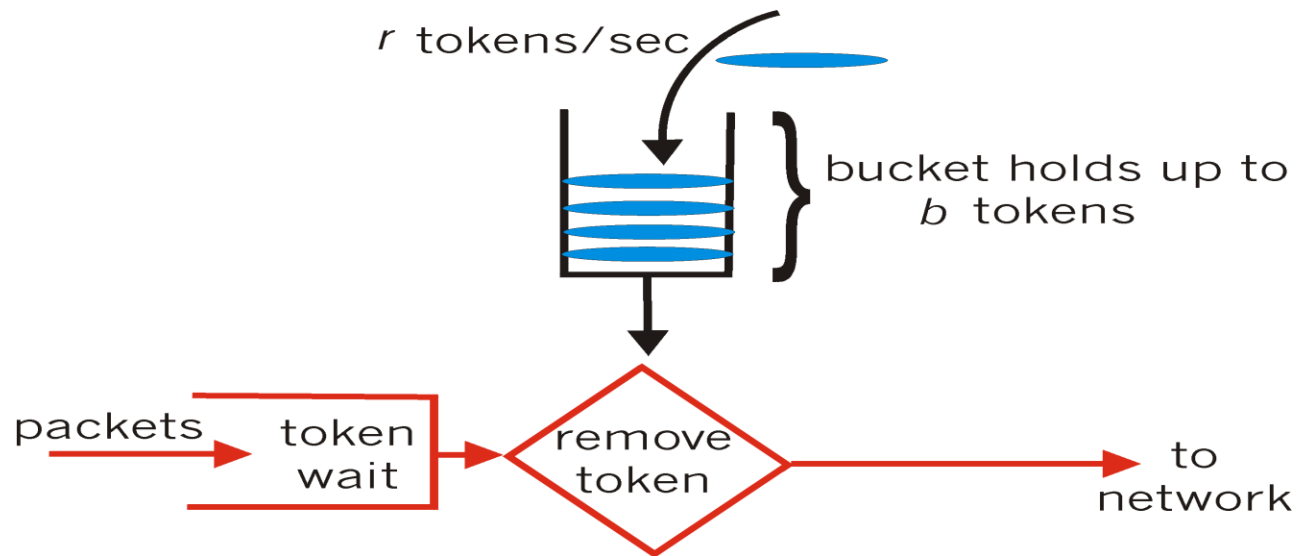
- ▶ Leaky bucket fuerza un patrón de tráfico de salida rígido (tasa promedio)
- ▶ Token bucket permite picos de tráfico (aumentar la velocidad , durante un intervalo pequeño) cuando le llega una ráfaga grande de paquetes
- ▶ *Aca los baldes (buckets) mantienen fichas (tokens), que son generadas por un clock a un rate de un token cada ΔT segundos*
- ▶ *Para transmitir, se necesita consumir un token. Si no hay , se espera*
- ▶ *Hosts Idle pueden “ahorrar” permisos para enviar bursts luego*

Token Bucket



“Policing Mechanisms”

- ▶ En definitiva Token Bucket permite un Burst Size y Average Rate.

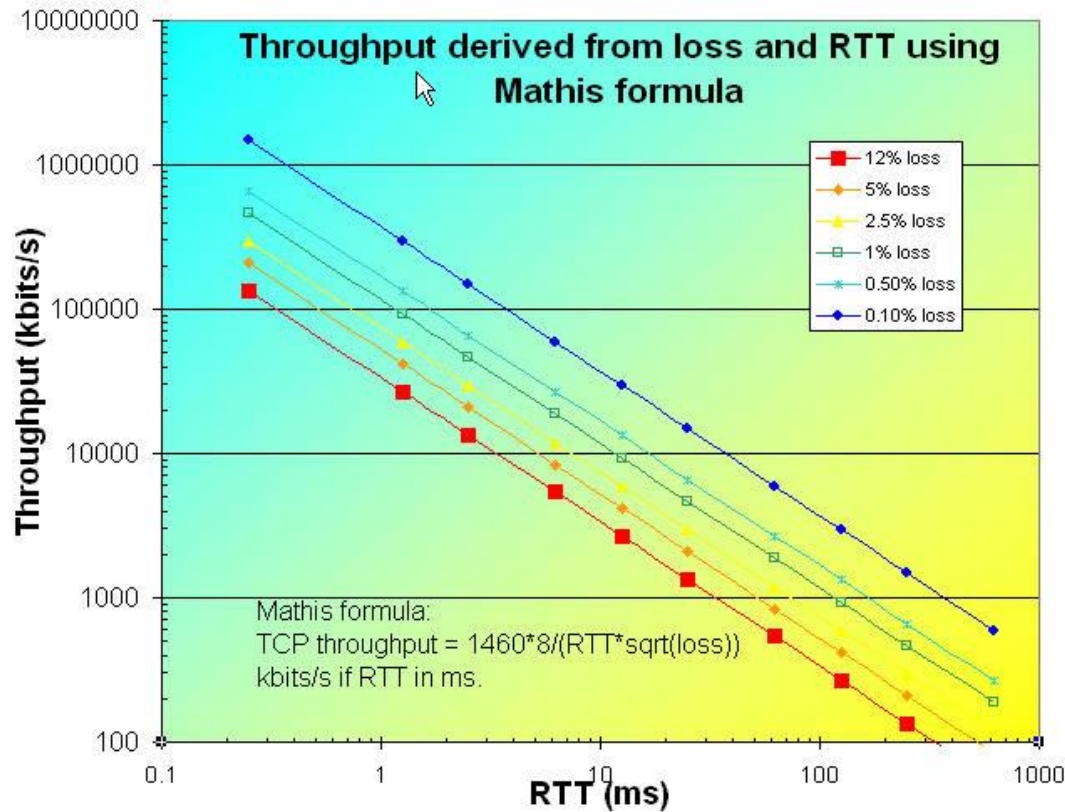


Para que puedo usar esos mecanismos:

- ▶ <http://geekis.me/uploads//2009/03/pbexperiment1.gif>



Performance TCP



$$BW = \frac{MSS * C}{RTT \sqrt{p}}$$

$$C = \sqrt{\frac{3}{2}}$$

Estrictamente en el RFC 2581 se define el Sender Maximum Segment Size (SMSS): es el máximo tamaño que el emisor de un segmento puede enviar en un momento dado. El valor puede estar acotado en el MTU (máxima unidad de transmisión) del enlace de salida desde el host, el resultado del Path MTU Discovery Algorithm [RFC1191], RMSS u otros factores. Este tamaño no incluye los encabezados TCP e IP.



Performance



Bufferbloat

- Agenda

- ▶ Presentar una primera aproximación al fenómeno de "*Bufferbloat*", básicamente como el tamaño del buffer en los dispositivos de red impacta en la latencia y por consiguiente en la performance de las aplicaciones. El aumento del tamaño de buffers puede llegar a mejorar el throughput de una sesión TCP pero a costa de un incremento de la latencia. De igual manera afectar a aplicaciones basadas en UDP.
- ▶ Como todos los dispositivos de red residenciales los fabricantes definen un tamaño estático de buffer (o dinámico en base algún algoritmo propietario), y en muchos casos dado el bajo costo del hardware de un tamaño excesivo (*overbuffering*), que termina generando una degradación de la performance del servicio de banda ancha.



OTROS OBJETIVOS

- ▶ Recordar la naturaleza “estadística” de las redes de conmutación de paquetes (Internet)
- ▶ El mecanismo de control de congestión de TCP como elemento fundamental en el funcionamiento de Internet
- ▶ La reciente especificación de DOCSIS 3.1 incluye nuevos algoritmos para la administración activa de los buffers , como también el desarrollo en openWRT en AP Wi-Fi



INTRODUCCIÓN : LA RED EN EL HOGAR DULCE HOGAR

iiiNo puedo jugar Pa !!!!!, Internet tiene Lag!!

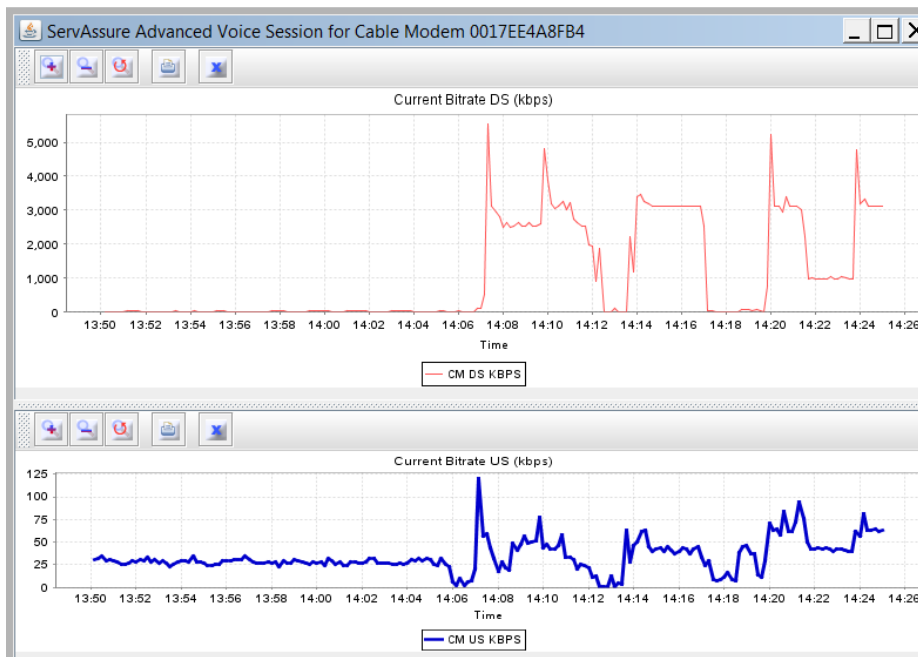


iiiiNico !!!!!, Larga youtube !!!!!!!



INTRODUCCIÓN : COMO LE EXPLICO ???

Ya se !: que TCP , sistemas de control de lazo cerrado y el problema de los buffer !!!



ANTECEDENTES (1)

- ▶ En 1985 John Nagle demuestra que el aumento del tamaño del buffer empeora el fenómeno de la congestión [1][2]
- ▶ En el 2009 David Reed (*) planteo que los problemas de alta latencia de la red 3G de AT&T se debían a un problema de diseño de la misma y no a falta de espectro. Atribuyo la causa al uso de buffers de gran tamaño en las RNCs (Radio Network Controller) para evitar las pérdidas de paquetes, que finalmente impacto en la latencia de la red [16] [17].

[*] Reed Entre otras cosas desarrollo el protocolo UDP

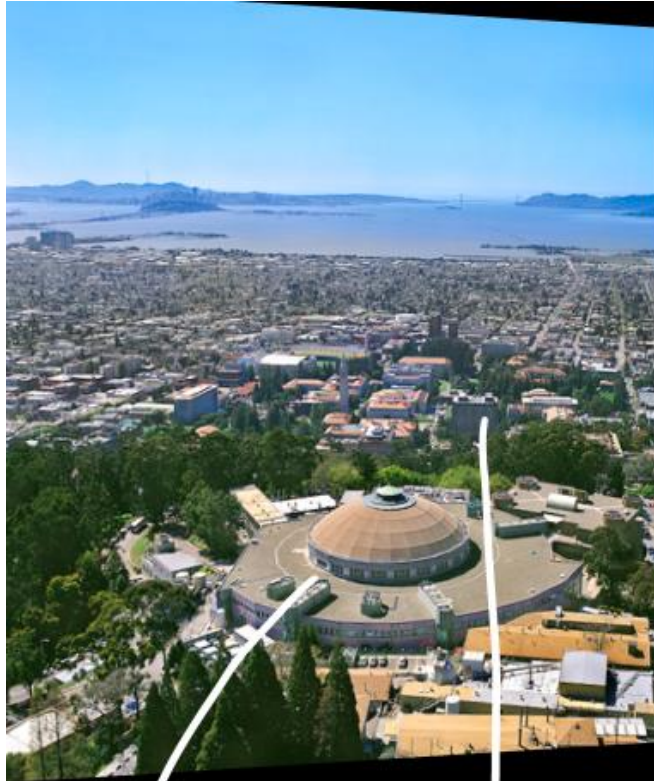


ANTECEDENTES (2)

- ▶ Comenzó a ser un problema nuevamente “conocido” a partir del trabajo de caracterización de Dischinger et. Al. [6].
- ▶ Los primeros en realizar una confirmación del excesivo buffereo presente en las redes en especial en los CPE y gateways hogareños fue el grupo ICSI de Berkeley mediante la herramienta Netalyzer [7].



INTERNET NO TENIA CONTROL DE CONGESTIÓN



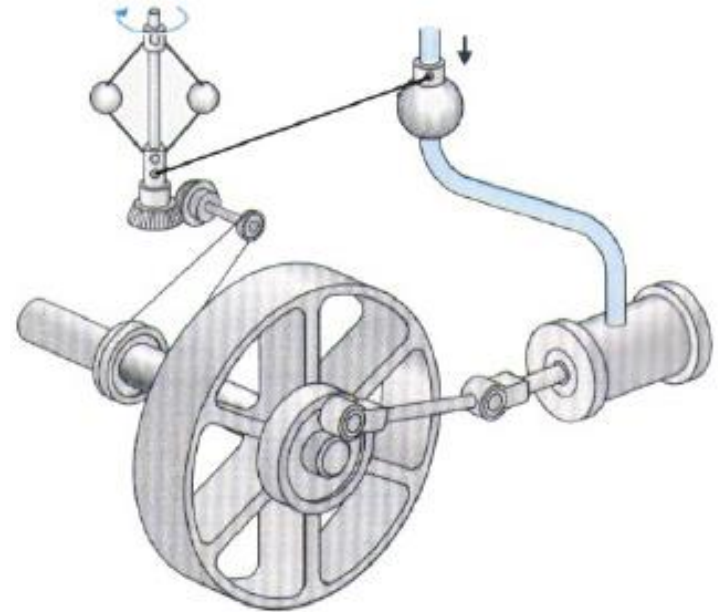
Lawrence Berkeley
National Laboratory

Electrical Engineering,
Berkeley University

"In October of '86, the Internet had the first of what became a series of 'congestion collapses'. During this period, the data throughput from LBL to UC Berkeley (sites separated by 400 yards and two IMP hops) dropped from 32 Kbps to 40 bps. We were fascinated by this sudden factor-of-thousand drop in bandwidth and embarked on an investigation of why things had gotten so bad."

Van Jacobson, "Congestion avoidance and control", 1988

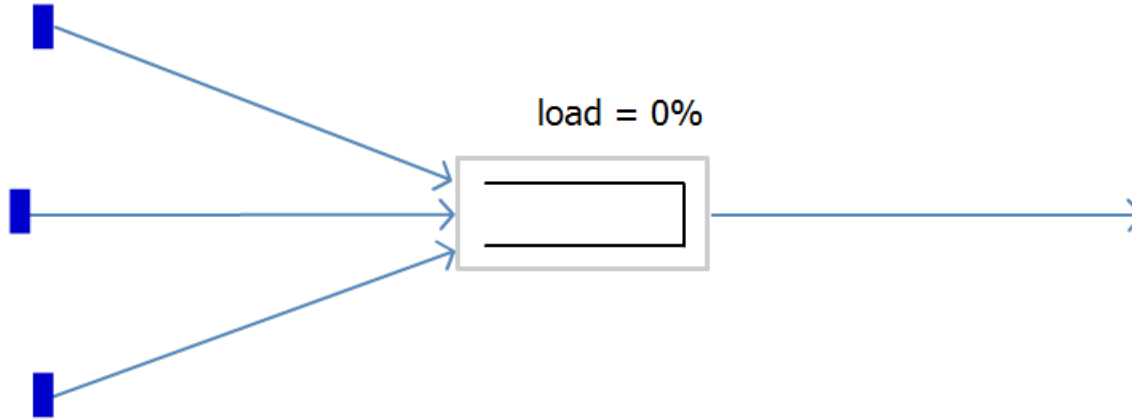
HASTA QUE LA “INVENTARON”



Dispositivo control presión en una
Máquina de vapor

de la presentación de Van Jacobson “Notes on Using Red for queue management
and Congestion Avoidance” Junio 1998

MULTIPLEXACIÓN ESTADÍSTICA

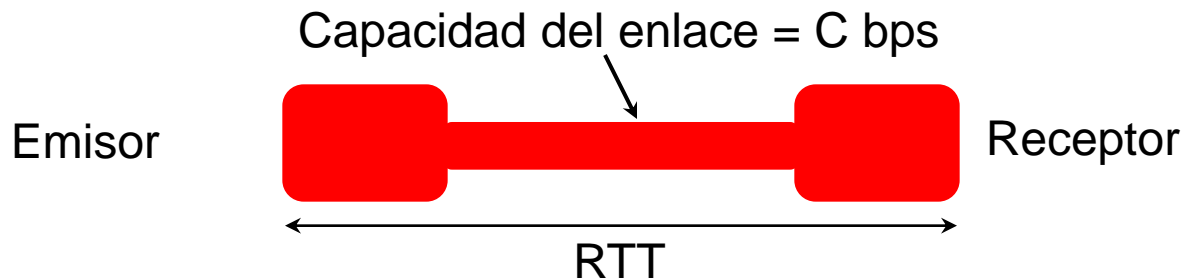


Observaciones

- ▶ Un buffer grande asegura perdida baja de paquetes
- ▶ Si el buffer nunca se vacía , ocupo el 100% del tiempo la línea de salida



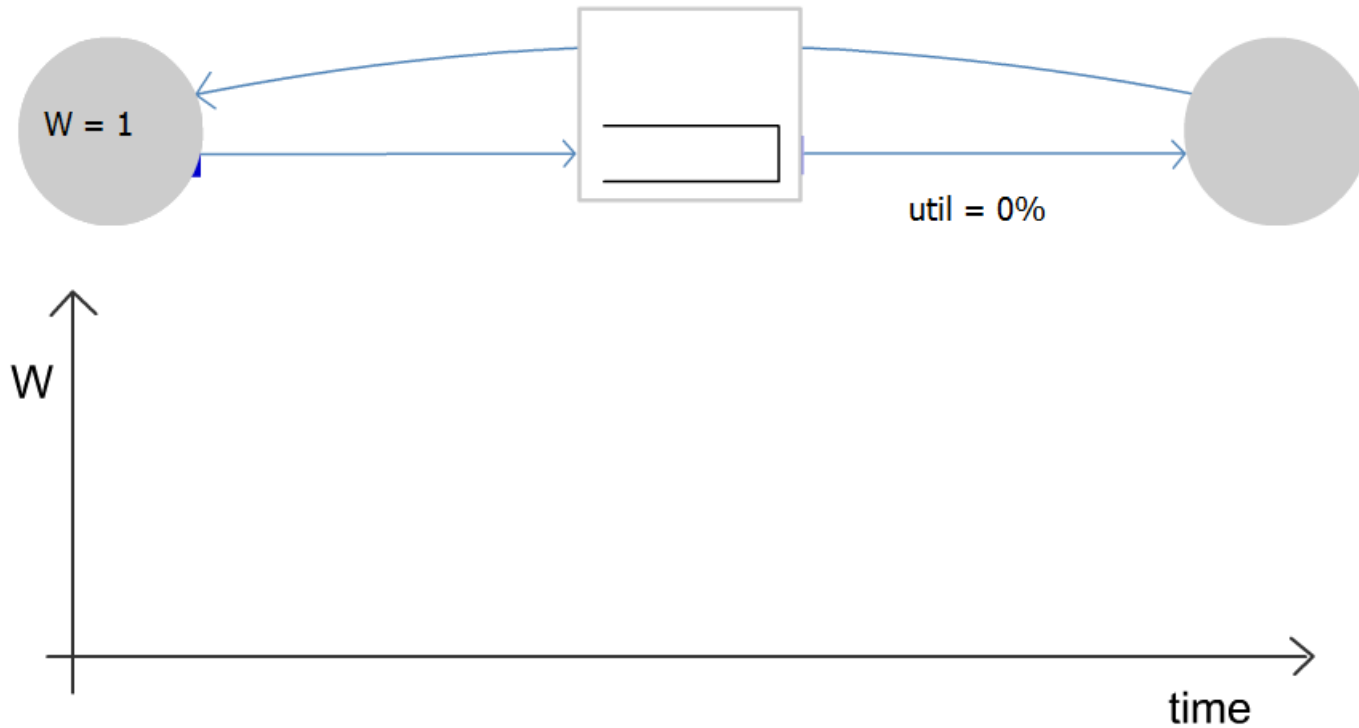
TCP CONTROL DE CONGESTION



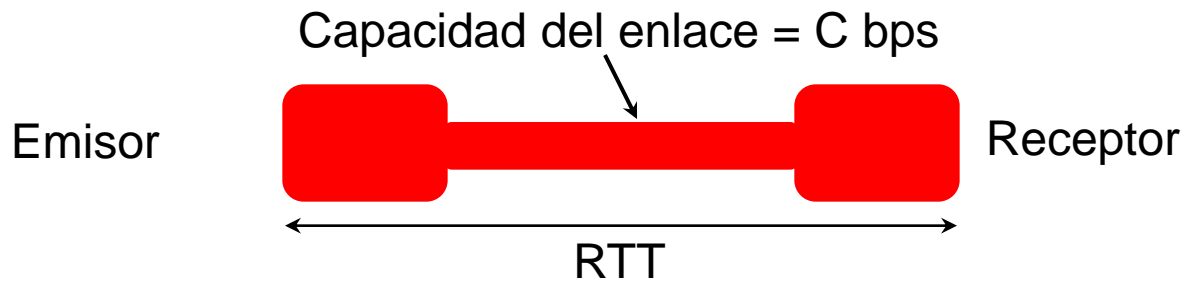
- TCP emisor “trata de llenar el pipe” incrementando la ventana W
- Idealmente crece W hasta $BDP = C \times RTT$
- TCP usa “packet loss” para detectar congestión y reducir la velocidad de emisión

TCP CONTROL DE CONGESTION

- ▶ Puedo tener solo " W " Paquetes "saliendo"
- ▶ W se ajusta en función del estado de la "red"
- ▶ Cuando llega un ACK se incrementa la ventana $W \leftarrow W + W/2$
- ▶ Si se pierde un paquete $W \leftarrow W/2$



VELOCIDAD DE UNA CONEXIÓN TCP: REGLA PRACTICA



- W Tamaño máximo 64 KiB [*]
- T Velocidad

$$T < W/RTT$$

- ¡Hay casos que si RTT es grande T no alcanza C!
- ¡ Para C muy grandes tampoco T alcanza ese valor !

¿A QUE SE LLAMA “BUFFERBLOAT” ?

- ▶ Es durante el 2011 que Jim Gettys de Alcatel-Lucent, que acuñó el termino Bufferbloat, comenzó a publicar diversos artículos que plantean que la “lentitud de Internet ” es a causa del *Bufferbloat* [3][4] además de presentaciones en el IETF [11], en Google Tech Talk [21] o en el congreso de operadores de redes de USA [24]
- ▶ Lo define formalmente como “la existencia de buffers excesivamente grandes en los sistemas, en particular en los relacionados con las redes y comunicaciones” [5]



Cable Modem Buffer Management in DOCSIS Networks

Jim Martin,
James Westall
School of Computing
Clemson University
Clemson, SC
jim.martin,westall
@cs.clemson.edu

Terry Shaw,
Greg White
CableLabs
Louisville, CO
t.shaw,g.white
@cablelabs.com

Rich Woundy
Software and Operations
Comcast Cable
Philadelphia, PA
richard_woundy
@cable.comcast.com

Jeff Finkelstein
Network Architecture
Cox Communications
Atlanta, GA
jeff.finkelstein
@cox.com

George Hart
Network Strategy
Rogers Communications
Toronto, Ontario
george.hart
@rci.rogers.com

Abstract—A critical component of subscriber management in a DOCSIS-based cable access network is the buffer management strategy that is in operation at the upstream service flow queue located in the cable modem. The strategy must contend with conflicting goals: large buffers might be required to ensure TCP flows can utilize available bandwidth, however large buffers can impact application flows that are latency sensitive. In this study, we have explored the relationship between application performance and upstream queue management. We define the optimal queue capacity as the buffer size that maximizes throughput and minimizes packet delay. Our results agree with previous work in wired Internet router contexts that suggest that the queue capacity should not exceed a bandwidth*delay product (BDP) amount of data. However, the upstream data rate available to a cable modem varies with the number of competing cable modems. Upstream service flow queues that are provisioned to hold a BDP of data might suffer queue delays exceeding several seconds during periods of congestion. Active queue management such as the Random

In spite of these advancements, there are rudimentary challenges that broadband cable service providers face. One such issue is subscriber bandwidth management. A crucial component to subscriber management in a DOCSIS-based cable access network is the buffer management strategy that is in operation at the upstream service flow queue located in the cable modem. While the upstream scheduling algorithm operating at the Cable Modem Termination System (CMTS) ensures quality-of-service commitments and determines the order that cable modems access the channel, the upstream buffer management strategy significantly impacts the performance of best effort applications during periods of congestion.

When a packet arrives at the cable modem from the subscriber's network destined for the Internet, the packet will be queued in an appropriate service flow queue. Assuming the packet is associated with a best-effort service, if a request for upstream bandwidth to the CMTS is not already in

BUFFERS EN DOCSIS

- ▶ Desde abril del 2011 para certificar CM DOCSIS 3.0 es mandatorio soportar **control de buffer [*]**. Los parámetros de control de buffer limitan la cantidad máxima de datos que pueden ser encolados por cada flujo de servicio. De esta manera Cablelabs estandariza [**] mediante la provisión de estas facilidades a los operadores un mecanismo para balancear la relación throughput y latencia de un servicio
- ▶ Los parámetros de control del buffer es un conjunto de tuplas Type-Length-Value (TLV) que definen un rango de tamaños aceptables de buffers. Permitiendo al CM/CMTS una cierta flexibilidad en la configuración del mismo.
- ▶ [*] Ver sección “C.2.2.5.11 Buffer Control” en [9] o en la revisión [10].
- ▶ [**] Los parámetros Buffer Control se expresan en bytes incluyen MAC frame data, MAC header HCS y el CRC para el MAC frames encolado para ese flujo de servicio. Este parámetro es aplicado al service flow.

BUFFERS EN DOCSIS

- ▶ Los parámetros de control del buffer es un conjunto de tuplas Type-Length-Value (TLV) que definen un rango de tamaños aceptables de buffers. Permitiendo al CM/CMTS una cierta flexibilidad en la configuración del mismo.
- ▶ **Minimum buffer:** define el límite inferior del tamaño de buffer para una flujo de servicio. Si el dispositivo no puede asignar esta cantidad mínima el servicio no es aceptado.
- ▶ **Maximum Buffer:** define el límite superior aceptable del buffer.
- ▶ **Target Buffer:** define el tamaño deseable de buffer. El dispositivo debe seleccionar el valor más cercano posible teniendo en cuenta su aplicación. Si se omite este parámetro el dispositivo selecciona cualquier valor de este parámetro dentro del rango aceptable, mediante un algoritmo específico del vendor.

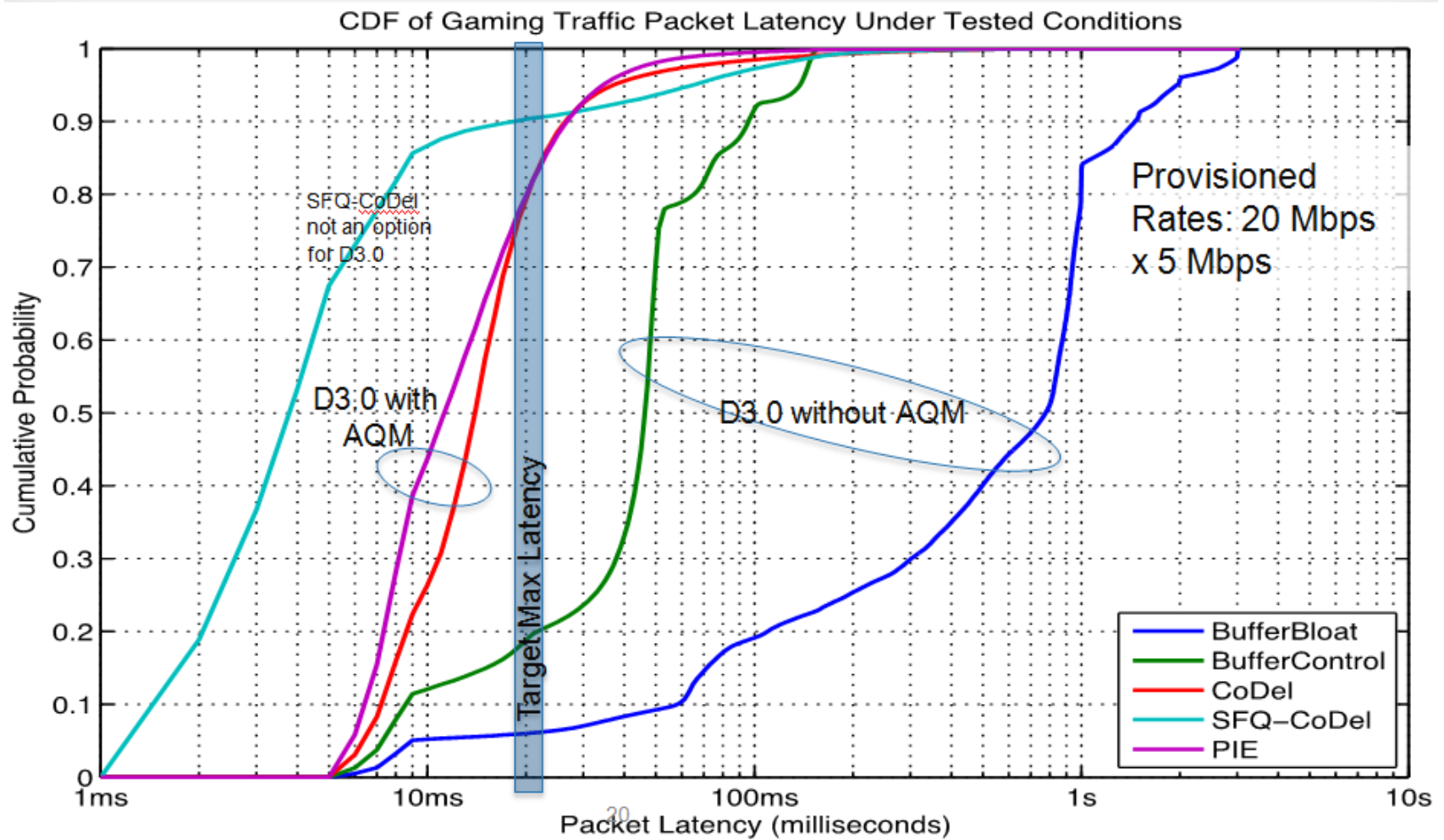


DIFERENTES PROPUESTAS EVALUADAS PARA DOCSIS

- ▶ Control de Buffer
- ▶ Algoritmos de AQM (Adaptive Queue Management) :
 - CoDel (Controlled Delay)
 - sfq_codel (Stochastic Flow Queue)
 - PIE (Proportional Integral Enhanced)

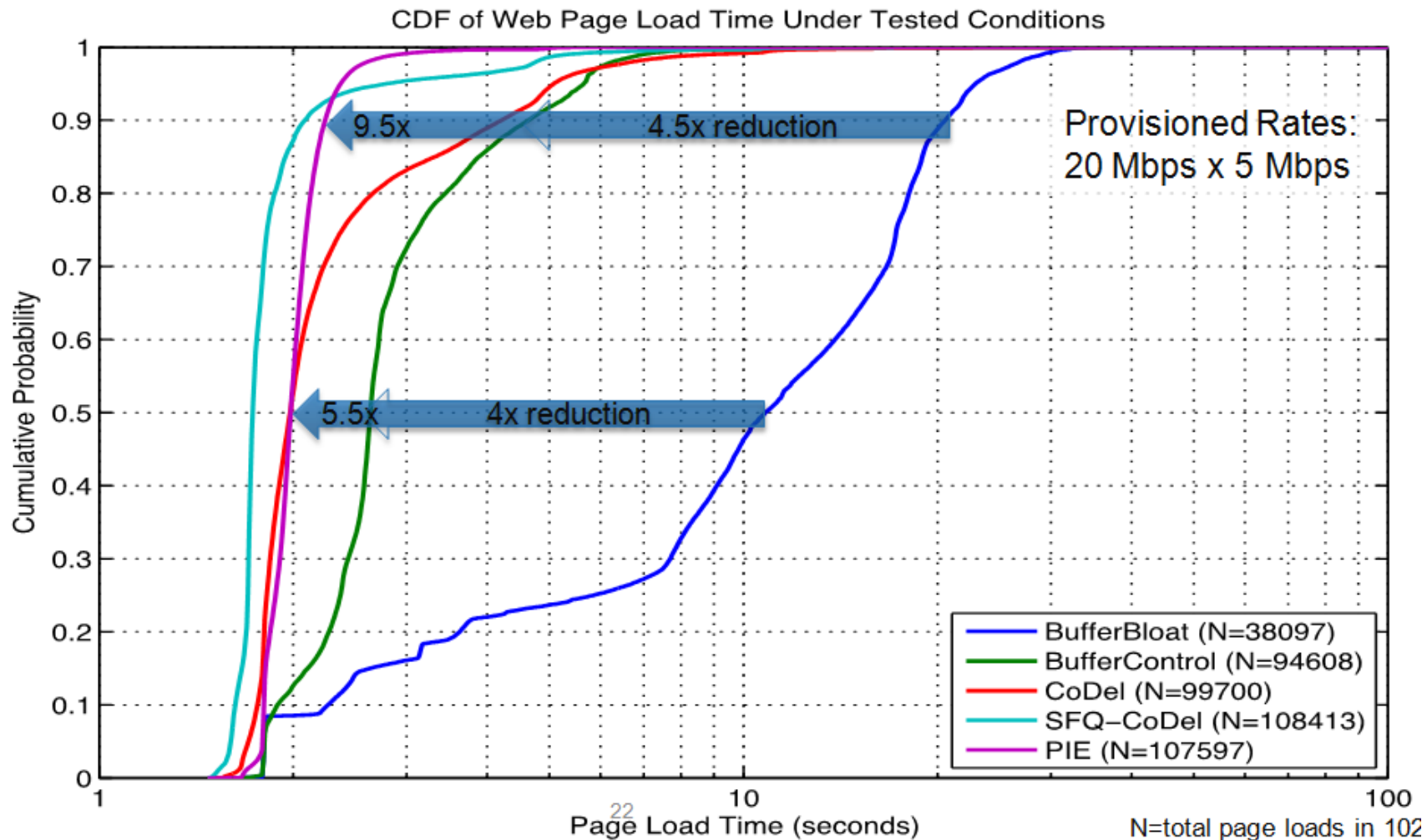


LATENCIA EN GAMING DOCSIS 3.0 [*]



[*] Fuente: Greg White , PhD. CableLabs

DEMORA EN LA CARGA DE UNA "PAGINA WEB DOCSIS" 3.0 [*]



[*] Fuente: Greg White, PhD. CableLabs